Leveraging K-Nearest Neighbors with SMOTE and Boosting Techniques for Data Imbalance and Accuracy Improvement

Adyanata Lubis[1,*], Yuda Irawan[2], Junadhi[3], Sarjon Defit[4]

[1]*Computer Science, Universitas Rokania, Pasir Pangaraian, Indonesia*

[2]*Computer Science, Universitas Hang Tuah Pekanbaru, Pekanbaru, Indonesia*

[3]*Computer Science, Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia*

[4]*Information Technology, Universitas Putra Indonesia YPTK, Padang, Indonesia*

(Received: August 15, 2024; Revised: September 14, 2024; Accepted: September 21, 2024; Available online: October 15, 2024)

Abstract

This research addresses the issue of low accuracy in sentiment analysis on Israeli products on social media, initially achieving only 64% using the K-NN algorithm. Given the ongoing Israeli-Palestinian conflict, which has garnered widespread international attention and strong opinions, understanding public sentiment towards Israeli products is crucial. To improve accuracy, the study employs SMOTE to handle data imbalance and combines K-NN with boosting algorithms like AdaBoost and XGBoost, which were selected for their effectiveness in improving model performance on imbalanced and complex datasets. AdaBoost was chosen for its ability to enhance model accuracy by focusing on misclassified instances, while XGBoost was selected for its efficiency and robustness in handling large datasets with multiple features. The research process includes data pre-processing (cleaning, normalization, tokenization, stopwords removal, and stemming), labeling using a Lexicon-Based approach, and feature extraction with CountVectorizer and TF-IDF. SMOTE was applied to oversample the minority class to match the number of instances in the majority class, ensuring balanced representation before model training. A total of 1,145 datasets were divided into training and testing data with a ratio of 70:30. Results demonstrate that SMOTE increased K-NN accuracy to 77%. Interestingly, combining K-NN with AdaBoost after SMOTE achieved 72% accuracy, which, although lower than the 77% achieved with SMOTE alone, was higher than the 68% accuracy without SMOTE. This discrepancy can be attributed to the added complexity introduced by AdaBoost, which may not synergize as effectively with SMOTE as XGBoost does, particularly in this dataset's context. In contrast, K-NN with XGBoost after SMOTE reached the highest accuracy of 88%, demonstrating a more effective combination. Boosting without SMOTE resulted in lower accuracies: 68% for KNN+AdaBoost and 64% for KNN+XGBoost. The combination of K-NN with SMOTE and XGBoost significantly improves model accuracy and reliability for sentiment analysis on social media.

Keywords: K-NN, XGBoost, AdaBoost, SMOTE, Machine Learning

1. Introduction

The Israeli-Palestinian conflict has long been a global issue, sparking controversy and strong emotions worldwide. Topics such as boycotting Israeli products frequently surface on various social media platforms, where users express their opinions, support, or protests concerning diverse social and political issues. Sentiment analysis provides valuable insights into public opinion, which can influence political and economic decisions. However, sentiment analysis in this context is challenging due to the complexity and imbalance of the data. Social media discussions on this topic often contain a mix of positive, negative, and neutral sentiments, with negative sentiments (e.g., calls for boycotts) frequently dominating. This imbalance can skew analysis results if not appropriately addressed.

In recent years the application of machine learning techniques to analyze sentiment on social media has emerged as a compelling research topic [1]. Given the vast volume of data and the intricate interactions among users, manual sentiment analysis has become increasingly impractical. Consequently, machine learning techniques like K-Nearest Neighbors (KNN) offer an effective solution for extracting sentiment patterns from social media data. To tackle these challenges, we employed several advanced technical methods. First, we used KNN, a robust algorithm known for its effectiveness in classification tasks. However, KNN alone can be less effective in highly imbalanced datasets, like those found in discussions of the Israeli-Palestinian conflict. The application of KNN in sentiment analysis offers a

***Corresponding author: Adyanata Lubis (adyanata@gmail.com)**

DOI: https://doi.org/10.47738/jads.v5i4.343

This is an open access article under the CC-BY license (https://creativecommons.org/licenses/by/4.0/).
© Authors retain all copyrights

straightforward yet powerful approach. The algorithm functions by comparing the data to be predicted with existing training data, then assigns a label based on the majority of the labels of its nearest neighbors. In the context of sentiment analysis, KNN can classify messages or comments on social media into specific sentiment categories, such as positive, negative, or neutral. The choice of KNN is motivated by its simplicity in implementation and interpretation, as well as its ability to perform well on small to medium-sized datasets.

While KNN offers the advantage of simplicity, it faces limitations in performance and scalability, particularly when dealing with large and diverse datasets. To overcome these limitations, integrating additional algorithms such as AdaBoost and XGBoost, along with data processing techniques like SMOTE (Synthetic Minority Over-Sampling Technique), can significantly enhance model accuracy and performance. These advanced methods help to address class imbalance and improve the predictive power of sentiment analysis models, making them more robust and effective in handling extensive social media data. AdaBoost (Adaptive Boosting) is a popular ensemble algorithm in machine learning. It works by combining several weak learners to form a strong learner. By assigning more weight to the errors made by the weak models, AdaBoost can enhance the model's performance in classifying data. The choice of AdaBoost is driven by its ability to improve model performance efficiently and scalably. This makes it a valuable addition to sentiment analysis, where it helps to create a more accurate and reliable classification system. Meanwhile XGBoost is a gradient boosting algorithm renowned for its ability to handle large and complex datasets and optimize various evaluation metrics such as accuracy and AUC (Area Under Curve). XGBoost was selected due to its reliability in managing intricate and extensive data, as well as its capacity to effectively fit models with diverse parameters.

The application of the SMOTE technique is crucial in addressing the issue of class imbalance within the dataset. In the context of sentiment analysis, there is often a disparity between the amounts of data with positive, negative, and neutral sentiments. SMOTE helps mitigate this issue by generating synthetic samples of the minority class, thus improving class balance and enhancing model performance. The reason for choosing SMOTE lies in its effectiveness in enhancing the representation of minority classes, which are often overlooked in imbalanced datasets.

By integrating machine learning algorithms such as KNN with AdaBoost, XGBoost, and SMOTE techniques, it is expected to improve the accuracy and reliability of the model in analyzing sentiment related to the relocation of the National Capital in Indonesia on social media platforms. Previous research supports the effectiveness of this method. For example, a study by wang [2] demonstrates that the integration of KNN in sentiment analysis on the Weibo platform can handle informal text effectively. Another study by pamuji [3] on the Instagram platform highlights the use of KNN to identify sentiment patterns in user posts. Additionally, research by mantik [4] using KNN in sentiment analysis for short video content on TikTok shows that this algorithm is adaptive to various types of data and platforms. Another study by mohammed [5] introduced the SMOTE technique and demonstrated improved model performance under conditions of extreme class imbalance. Research by natras [6] shows the effectiveness of AdaBoost in combining several weak models to form a stronger model. Meanwhile, ghosal [7] describe XGBoost's ability to optimize model accuracy and performance with large and complex datasets.

This research holds high relevance in understanding the dynamics of public opinion on sensitive and controversial issues on social media, particularly views on the boycott of Israeli products. The combination of the SMOTE technique with boosting algorithms such as AdaBoost and XGBoost is expected to provide significant improvements in the accuracy and reliability of sentiment analysis. This approach will offer deeper insights into public perceptions of the boycott of Israeli products. Previous research has demonstrated various approaches to addressing class imbalance issues in sentiment analysis and disease detection using machine learning algorithms. For instance, Govindarajan [8] showed that using Sequential Feature Selection (SFS) in the KNN algorithm can improve diabetes prediction accuracy by up to 2.6% by reducing computational complexity and selecting optimal features such as glucose levels and blood pressure. Meanwhile research by zamsuri [9] used KNN to classify various emotions in Indonesian text and found that the TF-IDF technique outperformed Bag-of-Words in terms of accuracy, achieving the best results of 79% for two-label classification. Additionally, research by nanda [10] which utilized Information Gain and KNN to detect fake news related to COVID-19 on Twitter based on the credibility of the author, achieved 91% accuracy. This underscores the effectiveness of combining these methods in the classification of social media data.

In another study elmogy [11] demonstrated that although various classifiers such as KNN, Naive Bayes, and Random Forest have been used for fake review detection, techniques to overcome data imbalance are still needed to enhance the performance of fake review detection. Research by barid [12] utilized the SMOTE technique to improve model accuracy in air quality classification. The results indicated that the SMOTE technique could significantly improve model accuracy by addressing data imbalance problems. Additional research by kasanah [13] shows that the use of SMOTE in machine learning algorithms can enhance the accuracy of online news objectivity classification at certain k values, although this technique sometimes decreases accuracy at higher k values. Another study by barid [12] demonstrated that the implementation of ensemble learning techniques with SMOTE on air quality data provides a significant increase in accuracy compared to conventional sampling methods. However, this method still faces challenges regarding parameter adjustment and optimal feature selection. Research by supriyanto [14], which evaluated the performance of machine learning models for Android malware detection, found that the KNN algorithm performs well in malware detection but still requires methods to overcome significant data imbalance. Additionally, tomar [15] focused on heart disease prediction using various machine learning algorithms. Their results showed that the use of optimization techniques and proper algorithm selection can enhance prediction performance.

Other research highlights that although various classification techniques have been used for Android malware detection, such as in the study by mishra [16], the problem of class imbalance remains a major challenge that reduces model performance. Another study examined fake review detection using various machine learning algorithms, including KNN, Naive Bayes, SVM, Logistic Regression, and Random Forest. The results show that KNN (with K=7) provides the best performance, achieving an f-score of 82.40% after incorporating user behavior features [11]. Further research focuses on XGBoost hyperparameter optimization for network intrusion detection using the 2018 CSE-CIC-IDS dataset. The results indicate that XGBoost with optimized hyperparameters achieves an ROC score of approximately 0.99, demonstrating optimal performance in detecting network attacks. This research underscores that hyperparameter optimization can significantly enhance model performance chimphlee [17].

Research by nur ghaniaviyanto [18] employs Word2Vec and KNN for online news topic classification in Indonesian. The results indicate that the combination of Word2Vec and KNN provides good accuracy in classifying news topics. However, the use of the SMOTE technique is necessary to address class imbalance in the news dataset, along with integration with boosting techniques to further enhance classification performance. Another study developed a favorite book prediction system using various machine learning algorithms. This study found that the performance of different algorithms can vary depending on the characteristics of the dataset. A notable gap that can be addressed is the application of SMOTE to handle class imbalance in the book recommendation dataset, combined with KNN and XGBoost to improve prediction accuracy [19]. The development of the model proposed in this study, integrating the SMOTE technique with boosting algorithms such as AdaBoost and XGBoost on KNN, is expected to address the existing gap by providing significant improvements in sentiment analysis accuracy on unbalanced datasets. This model not only leverages the power of SMOTE to balance the data but also optimizes classification performance through boosting techniques that have proven effective in previous studies. This approach is anticipated to make a significant contribution to understanding the dynamics of public opinion on sensitive issues on social media platforms. By integrating the SMOTE technique and boosting algorithms with KNN, the proposed model aims to overcome the shortcomings of previous studies, offering a better solution to the problem of data imbalance and enhancing accuracy and reliability in sentiment analysis.

2. Research Methodology

This research was conducted to improve the accuracy of the model in performing classification on the Boycott Israeli Products dataset. This research process includes several important stages which include data collection, pre-processing, lexicon based for labeling, feature extraction, data sharing, modeling, and evaluation. The methodology flowchart illustrated in figure 1 provides an overview of the entire process followed in this research:

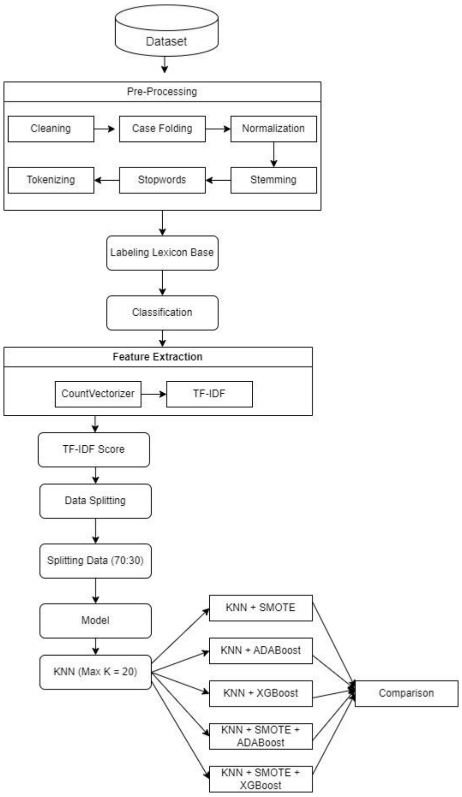


Figure 1. Proposed model

The following is a detailed explanation of each stage in the research method used:

2.1. Data Collection (Dataset)

The data for this study was collected from Twitter (now X) using a targeted crawling technique that focused on specific keywords related to the boycott of Israeli products. To ensure data quality and representativeness, several filtering criteria were applied during the data collection process. Only English-language tweets were included to maintain linguistic consistency. Spam tweets, identified by excessive links or repetitive content, were excluded. Data was gathered within a defined period from January 2023 to May 2024 to capture current sentiment. To enhance authenticity, tweets were selected from active users with at least 100 followers and regular activity, while retweets and duplicates were removed to ensure the dataset contained unique instances. These steps ensured the collected data was relevant, high-quality, and representative of genuine user sentiment.

2.2. Pre-Processing

The pre-processing stage is a crucial step in this research to ensure that the data used in the analysis is in optimal condition [20]. Pre-processing plays a vital role in sentiment analysis research. Data that has been cleaned and normalized improves model accuracy as the algorithm can focus on relevant features. Additionally, pre-processing simplifies and speeds up the modeling process because the data provided to the algorithm is already in a form ready for analysis [21]. Pre-processing is carried out through the following steps:

In the data cleaning stage, irrelevant elements such as special characters, punctuation marks, and numbers are removed to ensure only relevant data is processed further. URLs are eliminated using regular expressions, emojis are converted into text descriptions via the emoji library, hashtags are split into individual words, and user mentions are removed to maintain focus on the core textual content [22]. Next, all text is converted to lowercase to ensure consistency, avoiding unnecessary differences between words that are the same but written in different cases [23]. Normalization is performed using the 'key_norm.xlsx' dictionary, transforming word variations into standard forms to handle slang and local languages effectively, thereby improving the consistency and accuracy of text analysis [24]. In the tokenization stage, the text is broken down into individual words or tokens, which is essential for further analysis as machine learning algorithms require data in token form to analyze patterns and sentiments [25]. Stopwords, which are common words with little significance to the analysis, are then removed using the stopwords.xlsx dictionary combined with NLTK's stopword list, ensuring that irrelevant words from both colloquial and local languages are eliminated, thus reducing noise and enhancing analysis accuracy [26]. Finally, stemming reduces words to their base form by removing prefixes, suffixes, and infixes, which helps consolidate word variations, allowing the machine learning models to more efficiently identify patterns and improve sentiment analysis accuracy [27].

2.3. Labeling

After pre-processing, the next stage is data labeling using a Lexicon-Based approach to determine the sentiment polarity of each text. The lexicon used is based on words with predefined polarities for positive, negative, or neutral sentiments [28]. This method involves matching words in the text with those in the lexicon to assign sentiment labels, helping to categorize the data accurately according to its sentiment. While this method is simple and efficient for large datasets, it has limitations, such as its reliance on the completeness of the lexicon, its inability to understand context (e.g., "great disappointment" being labeled as positive), and its difficulty in detecting sarcasm or nuanced sentiments. To address these issues, the lexicon was customized with domain-specific terms; however, its inherent constraints remain. Despite these limitations, the lexicon-based approach is advantageous for its speed and simplicity in quick sentiment analysis.

2.4. Feature Extraction

At this stage, the features of the text are extracted using two main methods, CountVectorizer (Counts the frequency of occurrence of words in the document), and TF-IDF (Term Frequency - Inverse Document Frequency) to Calculates the importance weight of each word in a document based on the frequency of its occurrence and the number of documents containing that word [29]. These techniques were chosen for their simplicity, interpretability, and effectiveness in text classification tasks, particularly when dealing with sparse data. CountVectorizer and TF-IDF are well-suited for capturing the importance of words within documents and across the entire corpus, making them ideal for this sentiment analysis, where understanding term frequency and document relevance is crucial. Although advanced methods like word embeddings (e.g., Word2Vec, GloVe) and neural network-based approaches (e.g., BERT) offer more complex semantic representations, they also require significantly more computational resources and are prone to overfitting, especially in smaller datasets. Our focus was on creating a robust and interpretable model that balances performance with computational efficiency, making CountVectorizer and TF-IDF the preferred choice for this study.

2.5. Data Splitting

The data that has been processed and extracted for features is then divided into two parts: training data and test data, with a ratio of 70:30. This division aims to test the performance of the model to be built [30], [31]. The train-test split method is also computationally less intensive, which was advantageous given the large size of the dataset and the multiple models being tested. Moreover, the alternative approach used in this study still provides valuable insights into the model's performance.

2.6. Modeling

This research utilizes the KNN algorithm with a maximum K value of 20 as the foundation for building a classification model. KNN is an effective algorithm for classifying data based on similarity, where new data points are classified by identifying the majority class of their K nearest neighbors. For each new data point, the KNN algorithm identifies the K closest points from the training set and determines the class based on the majority class of those neighbors [32]. To address data imbalance and improve model accuracy, KNN is combined with several performance-enhancement techniques. First, SMOTE is applied to generate synthetic samples for minority classes, preventing bias towards the majority class. SMOTE increases the model's accuracy and sensitivity by allowing it to better recognize patterns in minority classes, making it the preferred method over ADASYN and Tomek links for handling class imbalance [33]. AdaBoost is also used to improve accuracy by focusing on errors made by the previous model, creating a more robust classifier [34]. XGBoost, known for its efficiency and regularization techniques to reduce overfitting, further enhances classification performance when combined with KNN [35].

Additionally, various combinations of these techniques are explored. KNN + SMOTE + AdaBoost addresses both data imbalance and accuracy, with SMOTE generating synthetic samples of minority classes while AdaBoost corrects classification errors. This combination increases sensitivity to minority classes and improves overall prediction accuracy. Similarly, KNN + SMOTE + XGBoost is designed to optimize performance on imbalanced data, as SMOTE balances the dataset and XGBoost boosts accuracy through regularization. By combining these techniques, the model can effectively handle complex datasets, ensuring a robust and reliable classification performance [35]. Developing this combined model involves several steps, namely calculating the distance between samples and determining nearest neighbors using the formula:

$$d(\mathrm{x},\mathrm{x_i}) = \sqrt{\sum_{j-1}^{m}(x_j - x_{ij})}\,2 \tag{1}$$

Where $d(x, xi)$ d(x, x i) is the Euclidean distance between x and xi, xj and xij are the jth feature values of x and xi respectively, and m is the number of features. Determine k nearest neighbors based on the calculated distance. Then determine the majority class of the k nearest neighbors:

$$\hat{\mathrm{y}} = \mathrm{modus}(\mathrm{y_{i1}}, \mathrm{y_{i2}}, \dots, \mathrm{y_{ik}}) \tag{2}$$

Where $\hat{y}$ is the classification label and the mode of $y_{i1}, y_{i2}, \dots, y_{ik}$ is the most frequent class among the k nearest neighbors.

To solve the class imbalance problem with SMOTE by randomly selecting two minority samples x_i *and* x_j so as to generate a new sample by interpolation:

$$\text{New Sample } = x_i + \wedge\,.\,(\,x_j - x_i) \tag{3}$$

Training using the Boosting technique by defining a function:

$$\mathrm{L}(0) = \sum_{\mathrm{i}-1}^{\mathrm{N}} \mathrm{l}(\mathrm{y_i}, \mathrm{y_i}) + \sum_{\mathrm{t}-1}^{\mathrm{T}} \omega\ (\mathrm{f_t}) \tag{4}$$

The $l(y_i, y_i)$ is the loss function, $\omega(ft)$ is the regulation function to avoid overfitting and y_i is the prediction of the model.

Furthermore, it defines a regulation function to control model complexity:

$$\omega\ (\mathrm{f_t}) = \mathrm{yT} + \frac{1}{2} \wedge \sum_{\mathrm{j}-1}^{\mathrm{T}} \mathrm{w_j^2} \tag{5}$$

Where γ and $\wedge$ are parameters that control the complexity of the model, T is the number of leaf nodes and wj is the weight of the leaf nodes. This will help improve the performance of the model in dealing with imbalanced data and ensure higher classification accuracy.

2.7. Comparison

In the comparison stage, a performance comparison will be conducted between the built models to determine the best model for classifying the sentiment of the dataset. Evaluation is very important to see the extent of the success of the research [36]. Model performance evaluation will utilize metrics such as accuracy, precision, recall, F1-score, and the Elbow Method. The Elbow Method helps determine the optimal K value for KNN, while the Confusion Matrix provides a detailed breakdown of the model's classification performance. These metrics offer a comprehensive assessment of the models, with accuracy measuring overall correctness, precision indicating the proportion of true positives among predicted positives, recall showing the ability to identify actual positive cases, and the F1-score balancing precision and recall. This comparison aims to identify the most accurate, reliable, and effective model for handling imbalanced data in sentiment analysis.

3. Result and Discussion

This section describes the results and discussions that aim to answer all questions in this study, namely the classification of public opinion about the boycott of Israeli products using the KNN algorithm on Twitter. The analysis includes the implementation of data preprocessing, feature extraction, and the application of the KNN algorithm along with enhancements using SMOTE, AdaBoost, and XGBoost techniques. The following is the result of the Preprocessing stage, which has been labeled using a Lexicon-Based approach. The graph of the labeling results can be seen in the following figure 2:

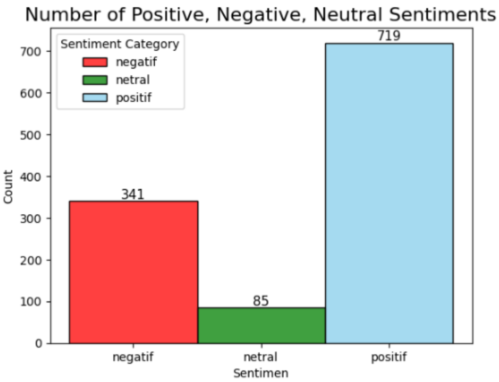
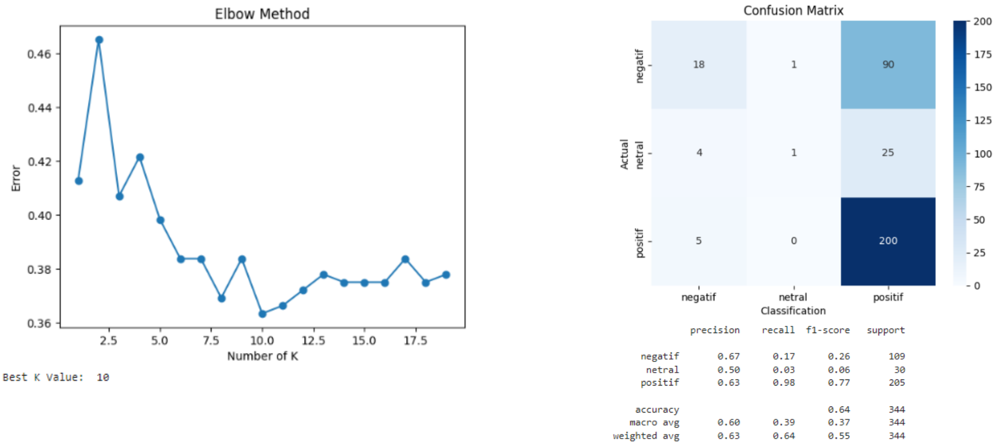


Figure 2. Lexicon-based Classification Results

The graph illustrates the distribution of sentiment labels (positive, negative, neutral) assigned to the dataset after the preprocessing and labeling stages. The graph shows a significant imbalance in the dataset, with 719 positive sentiments, 341 negative sentiments, and only 85 neutral sentiments. This imbalance can cause the machine learning model to be biased towards the majority (positive) class, this reducing the accuracy in classifying negative and neutral sentiments. From the labeling results, testing was carried out using the K-NN algorithm with the following results can be seen in the following figure 3:



```
Best K Value:  10
```

```
              precision    recall  f1-score   support

     negatif       0.67      0.17      0.26       109
      netral       0.50      0.03      0.06        30
     positif       0.63      0.98      0.77       205

    accuracy                           0.64       344
   macro avg       0.60      0.39      0.37       344
weighted avg       0.63      0.64      0.55       344
```

Figure 3. Elbow Method Graph and Confusion Matrix of KNN

The elbow method graph and the resulting confusion matrix show the performance of the K-NN algorithm that the error value decreases to the optimal point at a K value of about 10. However, the overall accuracy of the KNN model only reaches about 64%, which can be considered low. The confusion matrix shows that the KNN model performs very well in classifying positive sentiments (with 98% recall), but very poorly in classifying negative and neutral sentiments. This is shown by the high number of false negative and false neutral predictions. For example, out of 109 negative sentiment data, only 18 were correctly predicted, while the rest were predicted as positive sentiments. The cause of low accuracy is the problem of data imbalance. The dataset used in this study has a significant imbalance, with a much larger number of positive sentiments compared to negative and neutral sentiments. This imbalance makes the model more likely to predict the majority class (positive) more accurately, but less effective in predicting the minority class (negative and neutral). To overcome this problem this research uses the SMOTE technique to balance the data. In addition, the KNN algorithm is combined with boosting techniques such as AdaBoost and XGBoost to further improve classification performance. The results of this combination show a significant increase in accuracy, highlighting the importance of handling data imbalance and the use of ensemble techniques in sentiment analysis. With this approach the model becomes more effective in classifying all sentiment categories including previously underrepresented minority classes. The results of applying the SMOTE technique can be seen in figure 4 as follows:

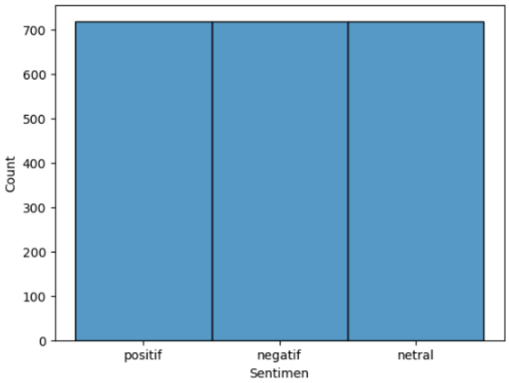
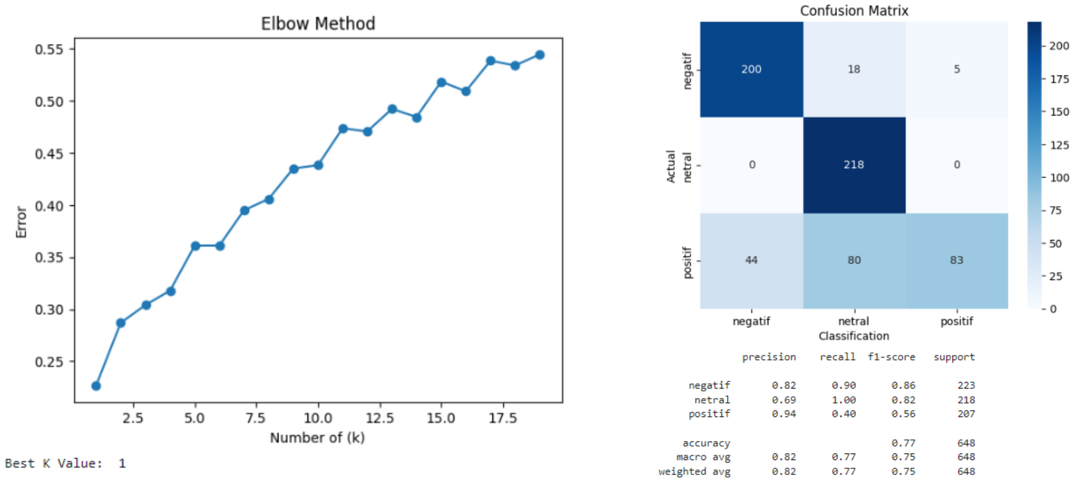


Figure 4. Data Balancing Results Using SMOTE

Figure 4 above shows that the SMOTE technique successfully creates a balance between the three sentiment categories (positive, negative, and neutral), each with around 700 data points. This balance is important to ensure that the KNN machine learning model is not biased towards one particular category, thus improving the overall performance of the model in classifying all sentiment categories more accurately. With balanced data, evaluation metrics such as accuracy, precision, recall, and f1-score will be more representative and unbiased, hopefully resulting in more reliable predictions. The next step is to retrain the model with this balanced dataset and monitor the evaluation metrics to ensure that the application of SMOTE provides a significant improvement in model performance. The results of the evaluation using the confusion matrix can be seen in figure 5 below:



```
Best K Value:  1
```

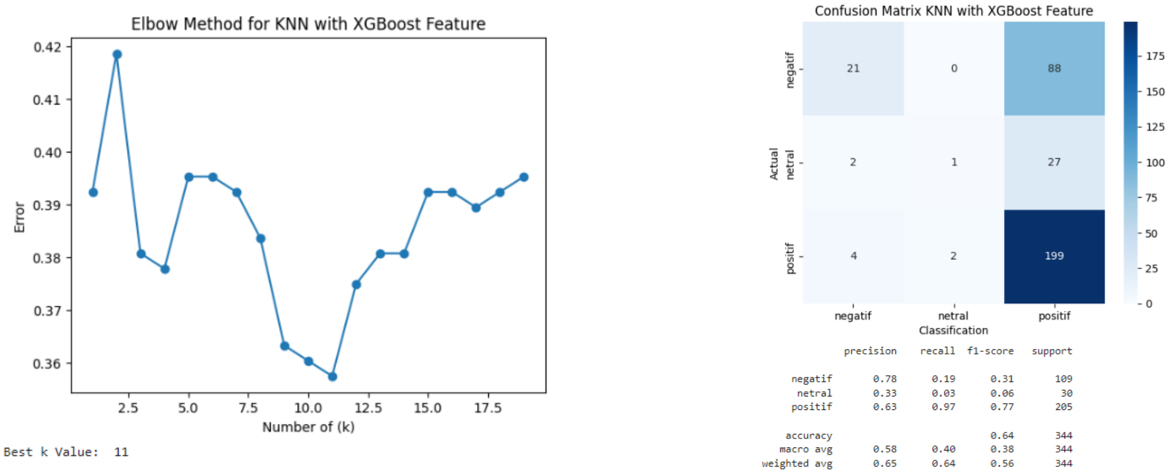
```
              precision    recall  f1-score   support

     negatif       0.82      0.90      0.86       223
      netral       0.69      1.00      0.82       218
     positif       0.94      0.40      0.56       207

    accuracy                           0.77       648
   macro avg       0.82      0.77      0.75       648
weighted avg       0.82      0.77      0.75       648
```

Figure 5. Elbow Method Graph and Confusion Matrix of KNN+SMOTE

After applying the SMOTE technique, it can be seen that the data distribution becomes more balanced which is reflected in the confusion matrix and elbow graph. From the confusion matrix, the KNN model with SMOTE shows better performance in classifying negative and neutral sentiments compared to before. Precision and recall for negative and neutral sentiment increased significantly, with recall for neutral sentiment reaching 100%. However, the model still shows weakness in classifying positive sentiments with recall only reaching 40%, indicating that the model still has difficulty in recognizing all categories equally. Compared to the results before the application of SMOTE, there is a clear improvement in the overall accuracy of the model from 63.66% to 77.31%. Before SMOTE, the model was heavily biased towards positive sentiments and failed to recognize negative and neutral sentiments well. With the application of SMOTE, although there is a significant improvement in accuracy and performance for negative and neutral sentiments, the model still needs to be further optimized to improve performance on positive sentiments. This shows that while SMOTE is effective in balancing the data, there needs to be additional customization or combination with other techniques to achieve optimal performance across all sentiment categories.

To further improve the performance of the model, this research also combines KNN with boosting techniques such as AdaBoost and XGBoost. This combination is expected to utilize the advantages of each technique, where AdaBoost can improve model accuracy by focusing on data that is difficult to classify, while XGBoost is known for its ability to handle large and complex datasets with high performance. The following is figure 6 and figure 7 which shows the test results using KNN-ADABoost and XGBoost without SMOTE:

KNN + XGBoost

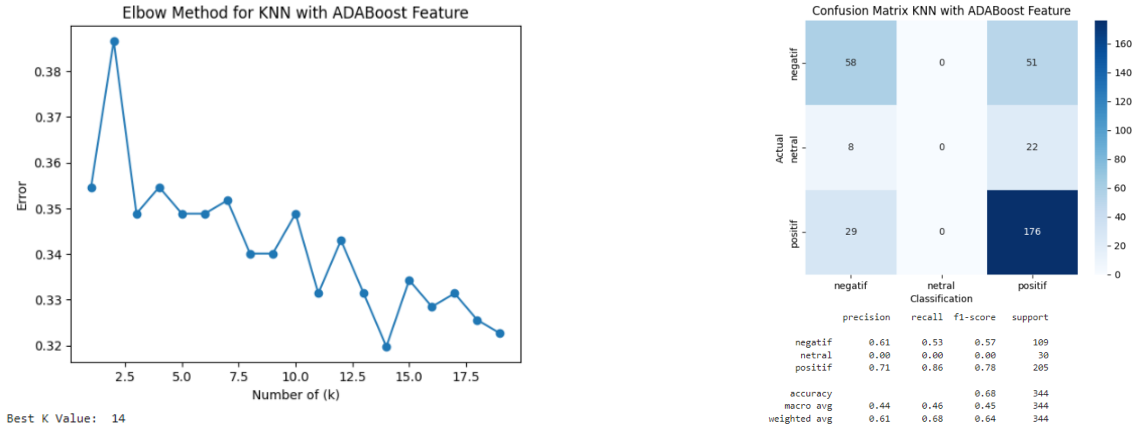


```
Best k Value:  11
```

```
              precision    recall  f1-score   support

     negatif       0.78      0.19      0.31       109
      netral       0.33      0.03      0.06        30
     positif       0.63      0.97      0.77       205

    accuracy                           0.64       344
   macro avg       0.58      0.40      0.38       344
weighted avg       0.65      0.64      0.56       344
```

Figure 6. Graph of Elbow Method and Confusion Matrix of KNN+XGBoost

KNN + ADABoost



```
Best K Value:  14
```

```
              precision    recall  f1-score   support

     negatif       0.61      0.53      0.57       109
      netral       0.00      0.00      0.00        30
     positif       0.71      0.86      0.78       205

    accuracy                           0.68       344
   macro avg       0.44      0.46      0.45       344
weighted avg       0.61      0.68      0.64       344
```

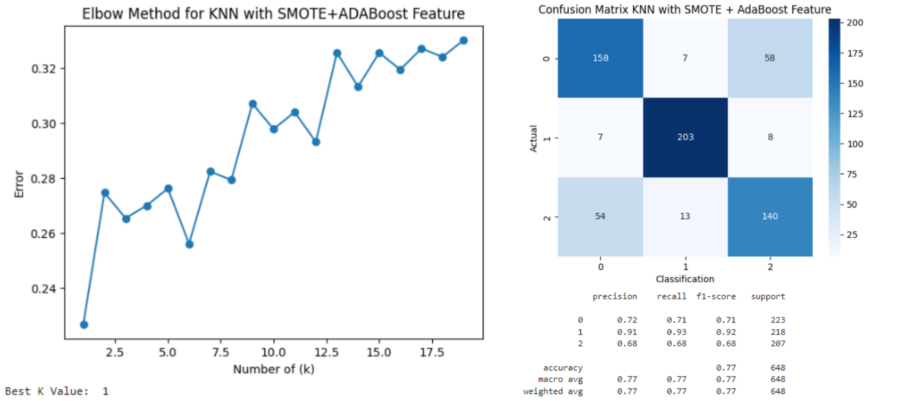
Figure 7. Graph of Elbow Method and Confusion Matrix of KNN+ADABoost

In the combination of KNN + XGBoost, the Elbow Method graph indicates that the optimal k value is 11, where the error is the lowest compared to other k values. The confusion matrix reveals that the KNN model with XGBoost features achieves an accuracy rate of 0.64. From the table below the confusion matrix, we observe that the highest precision is attained in the positive class with a value of 0.63 and a recall of 0.97, indicating the model's excellent performance in identifying positive instances. However, the negative class has a high precision of 0.78 but a low recall of 0.19, suggesting that many negative instances are not correctly identified. Overall, the weighted average for precision is 0.65, and for recall, it is 0.64.

In the KNN + ADABoost combination, the Elbow Method graph shows that the optimal k value is 14, with the lowest error at this point. The confusion matrix demonstrates that the KNN model with ADABoost features achieves an accuracy rate of 0.68. According to the table below the confusion matrix, the highest precision is achieved in the positive class with a value of 0.71 and a recall of 0.86. However, the neutral class shows extremely poor performance, with both precision and recall at 0.00. The negative class has a precision of 0.61 and a recall of 0.53, indicating better performance compared to the KNN model with XGBoost features for the negative class. Overall, the weighted average for precision is 0.61, and for recall, it is 0.68.

The KNN model with ADABoost features achieves a higher accuracy (0.68) compared to the KNN model with XGBoost features (0.64). However, the performance in the neutral class is very poor for the ADABoost model, whereas the XGBoost model shows a more balanced performance despite the very low recall in the negative class. The ADABoost model demonstrates a more stable error reduction in the Elbow Method graph compared to the XGBoost model. Both models illustrate that boosting features can enhance the performance of KNN, yet the ADABoost model appears more stable in reducing error at higher optimal k values. To optimize the performance of the model in accurately classifying sentiment, this research will conduct tests by combining KNN with SMOTE, XGBoost, and AdaBoost with the test results shown in the following figure 8 and figure 9:

KNN+SMOTE+ADABoost

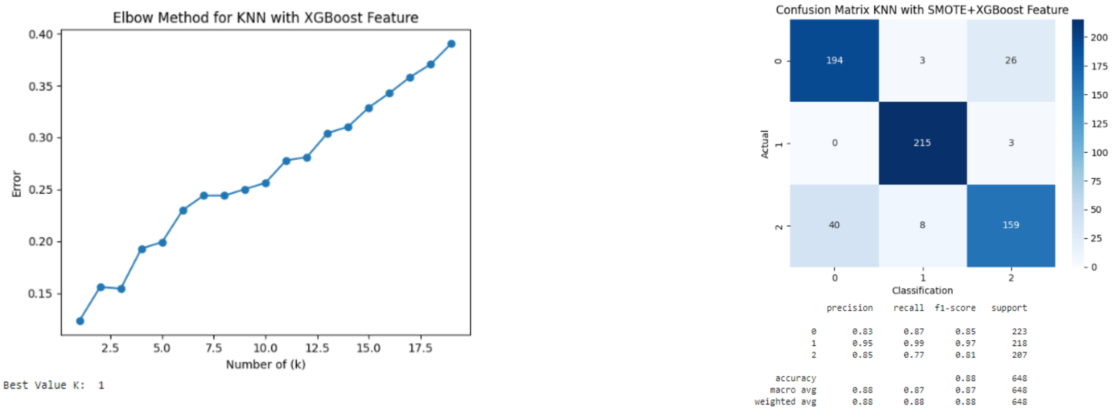


```
Best K Value:  1
```

```
              precision    recall  f1-score   support

           0       0.72      0.71      0.71       223
           1       0.91      0.93      0.92       218
           2       0.68      0.68      0.68       207

    accuracy                           0.77       648
   macro avg       0.77      0.77      0.77       648
weighted avg       0.77      0.77      0.77       648
```

Figure 8. Elbow Method Graph and Confusion Matrix of KNN+SMOTE+ADABoost

KNN+SMOTE+XGBoost

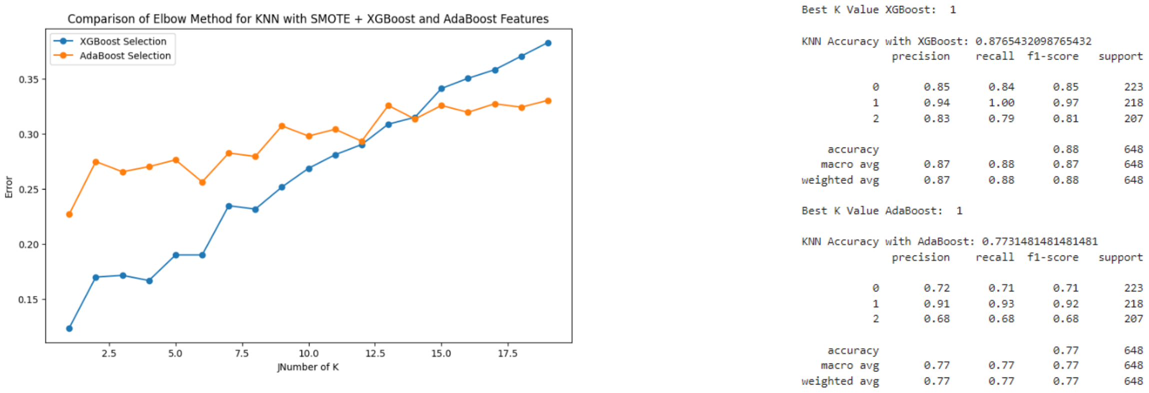


```
Best Value K:  1
```

```
              precision    recall  f1-score   support

           0       0.83      0.87      0.85       223
           1       0.95      0.99      0.97       218
           2       0.85      0.77      0.81       207

    accuracy                           0.88       648
   macro avg       0.88      0.87      0.87       648
weighted avg       0.88      0.88      0.88       648
```

Figure 9. Elbow Method Graph and Confusion Matrix of KNN+SMOTE+XGBoost

In the KNN+SMOTE+ADABoost combination, the Elbow Method graph indicates that the optimal k value is 1, where the error is the lowest. The confusion matrix reveals that the KNN model with SMOTE and ADABoost features achieves an accuracy of 0.77. From the table below the confusion matrix, it is observed that the highest precision is achieved in class 1 with a value of 0.91 and a recall of 0.93. However, class 2 shows lower performance with a precision of 0.68 and a recall of 0.68. Class 0 has a precision of 0.72 and a recall of 0.71, indicating balanced performance. Overall, the weighted average for precision is 0.77, and for recall, it is also 0.77. In the KNN+SMOTE+XGBoost combination, the Elbow Method graph similarly shows that the optimal k value is 1, with the lowest error at this point. The confusion matrix demonstrates that the KNN model with SMOTE and XGBoost features achieves a higher accuracy of 0.88. According to the table below the confusion matrix, the highest precision is attained in class 1 with a value of 0.95 and a recall of 0.99. Class 2 also performs well with a precision of 0.85 and a recall of 0.77. Class 0 shows the highest precision at 0.83 and a recall of 0.87. Overall, the weighted average for precision is 0.88, and for recall, it is also 0.88. Here is figure 10 which shows the comparison results between XGBoost and ADABoost:



```
Best K Value XGBoost:  1

KNN Accuracy with XGBoost: 0.8765432098765432
              precision    recall  f1-score   support

           0       0.85      0.84      0.85       223
           1       0.94      1.00      0.97       218
           2       0.83      0.79      0.81       207

    accuracy                           0.88       648
   macro avg       0.87      0.88      0.87       648
weighted avg       0.87      0.88      0.88       648

Best K Value AdaBoost:  1

KNN Accuracy with AdaBoost: 0.7731481481481481
              precision    recall  f1-score   support

           0       0.72      0.71      0.71       223
           1       0.91      0.93      0.92       218
           2       0.68      0.68      0.68       207

    accuracy                           0.77       648
   macro avg       0.77      0.77      0.77       648
weighted avg       0.77      0.77      0.77       648
```

Figure 10. Comparison of Elbow Method for KNN with SMOTE + XGBoost and AdaBoost Features

In this study, both XGBoost and AdaBoost were employed to enhance the performance of the K-NN algorithm. We performed parameter tuning to optimize the performance of these models, ensuring that they were tailored to the specific characteristics of our dataset. For AdaBoost, the primary parameters adjusted were the number of estimators and the learning rate. We experimented with a range of values for the number of estimators, testing between 50 to 200 estimators, and selected a learning rate between 0.01 and 1.0. The final model used 100 estimators and a learning rate of 0.1, as these settings provided the best trade-off between model complexity and accuracy. For XGBoost, we focused on tuning several key parameters: the maximum depth of the trees, the learning rate, and the number of boosting rounds. We tested maximum depths ranging from 3 to 10 and learning rates from 0.01 to 0.3. Additionally, we adjusted the number of boosting rounds between 100 to 500 rounds. The optimal configuration was found with a maximum depth of 6, a learning rate of 0.1, and 300 boosting rounds, which balanced model performance and computational efficiency. This tuning process was performed using grid search with cross-validation to ensure the robustness of the results. By carefully adjusting these parameters, we aimed to maximize the model's predictive power while preventing overfitting, particularly given the imbalanced nature of the dataset. The final tuned models were used for the performance evaluations presented in this study.

The figure presents a comparison of the Elbow Method for KNN combined with SMOTE and the features of XGBoost and ADABoost. In the KNN+SMOTE+XGBoost model, the optimal k value is 1, with an accuracy of 0.88. This model demonstrates the highest precision in class 1 with a value of 0.94 and a recall of 1.00, with weighted averages for precision, recall, and f1-score of 0.87, 0.88, and 0.88, respectively. Conversely, in the KNN+SMOTE+ADABoost model, the optimal k value is also 1, with an accuracy of 0.77. The highest precision in this model is achieved in class 1 with a value of 0.91 and a recall of 0.93, with weighted averages for precision, recall, and f1-score of 0.77. Although both models benefit from the application of SMOTE to address class imbalance, the KNN model with XGBoost features significantly outperforms the ADABoost-enhanced model in terms of accuracy, precision, and recall, making it the superior choice for applications requiring high performance across multiple classes. The following is figure 11 which shows a comparison graph of the accuracy of the models that have been tested:

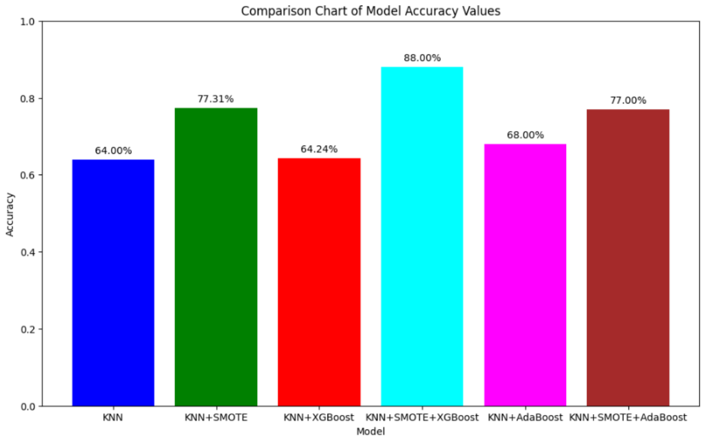


Figure 11. Comparison of Model Accuracy Values

The bar chart presents a comparative analysis of various KNN models, revealing that the KNN+SMOTE+XGBoost model achieves the highest accuracy at 88.00%, significantly outperforming other combinations. This result underscores the effectiveness of integrating SMOTE for handling class imbalance and leveraging XGBoost's boosting capabilities. The KNN+SMOTE model follows with an accuracy of 77.31%, indicating substantial benefits from SMOTE alone. Conversely, the KNN+ADABoost and KNN+SMOTE+ADABoost models achieve accuracies of 68.00% and 77.00%, respectively, suggesting that while ADABoost enhances performance, it is less effective than XGBoost in this context. The base KNN model and KNN+XGBoost show similar, lower accuracies (64.00% and 64.24%), highlighting the limited impact of XGBoost without addressing class imbalance. Thus, the integration of SMOTE and XGBoost emerges as the superior approach, offering significant accuracy improvements and making it the optimal choice for high-performance applications. In order to assess the effectiveness of the proposed methods, we compared the performance of our enhanced models (K-NN combined with SMOTE, AdaBoost, and XGBoost) against baseline models. The baseline used in this study was derived from the work of AG Pertiwi [37], who utilized a combination of K-NN and SMOTE for sentiment analysis, achieving an accuracy of 85%. Our enhanced models demonstrated significant improvements over this baseline. Specifically, the combination of K-NN with SMOTE and XGBoost achieved an accuracy of 88%, surpassing the 85% accuracy reported by Pertiwi.

4. Conclusion

The analysis results show that the use of SMOTE technique to balance the data significantly improves the performance of KNN model in sentiment classification. The combination of KNN with SMOTE and XGBoost produces the highest accuracy of 88%, compared to the combination of KNN with SMOTE and AdaBoost which is 77%. This shows that XGBoost is more effective in improving the accuracy and reliability of model predictions after data balancing. On the other hand, AdaBoost also improves model performance compared to without SMOTE, but not as effective as XGBoost. Without SMOTE, the KNN model showed significant bias towards the majority class and low performance in classifying the minority class with an accuracy of only 64%. When only using boosting techniques without SMOTE, although there is an increase in accuracy, the results are still not optimal because the data imbalance remains. The combination of KNN+AdaBoost without SMOTE achieved 68% accuracy, while KNN+XGBoost without SMOTE achieved 64%. This confirms the importance of data balancing with SMOTE before applying boosting techniques to achieve better model performance. For further development, it is recommended to perform more in-depth parameter tuning for XGBoost and AdaBoost and consider the use of other ensemble techniques such as Random Forest or Gradient Boosting Machines (GBM). In addition, integration of feature selection methods and more effective use of cross-validation can help improve accuracy. Consider combining more than one model by using ensemble learning techniques such as stacking, so as to improve model performance by utilizing the advantages of each algorithm.

5. Declarations

5.1. Author Contributions

Conceptualization: A.L., Y.I., J., and S.D.; Methodology: Y.I.; Software: J.; Validation: A.L. and Y.I.; Formal Analysis: A.L. and J.; Investigation: J.; Resources: S.D.; Data Curation: S.D.; Writing Original Draft Preparation: A.L., Y.I., J., and S.D.; Writing Review and Editing: A.L., Y.I., J., and S.D.; Visualization: A.L. and J. All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Herianto, B. Kurniawan, Z. H. Hartomi, Y. Irawan, and M. K. Anam, "Machine Learning Algorithm Optimization using Stacking Technique for Graduation Prediction," *J. Appl. Data Sci.*, vol. 5, no. 3, pp. 1272–1285, 2024.

[2] P. Wang, H. Shi, X. Wu, and L. Jiao, "Sentiment analysis of rumor spread amid covid-19: Based on weibo text," *Healthc.*, vol. 9, no. 10, 2021, doi: 10.3390/healthcare9101275.

[3] A. Pamuji, "Performance of the K-Nearest Neighbors Method on Analysis of Social Media Sentiment," *Juisi*, vol. 07, no. 01, pp. 32–37, 2021.

[4] R. D. A. Lestari, B. S. Rintyarna, and M. Dasuki, "Application Of N-Gram On K-Nearest Neighbor Algorithm To Sentiment Analysis Of TikTok Shop Shopping Features," *J. Mantik*, vol. 6, no. 3, pp. 2685–4236, 2022.

[5] A. J. Mohammed, "Improving Classification Performance for a Novel Imbalanced Medical Dataset using SMOTE Method," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 3161–3172, 2020, doi: 10.30534/ijatcse/2020/104932020.

[6] R. Natras, B. Soja, and M. Schmidt, "Ensemble Machine Learning of Random Forest, AdaBoost and XGBoost for Vertical Total Electron Content Forecasting," *Remote Sens.*, vol. 14, no. 15, pp. 1–34, 2022, doi: 10.3390/rs14153547.

[7] S. Ghosal and A. Jain, "Depression and Suicide Risk Detection on Social Media using fastText Embedding and XGBoost Classifier," *Procedia Comput. Sci.*, vol. 218, no. march, pp. 1631–1639, 2022, doi: 10.1016/j.procs.2023.01.141.

[8] R. Govindarajan, V. Balaji, J. Arumugam, T. A. Assegie, and R. Mothukuri, "Evaluation of sequential feature selection in improving the K-nearest neighbor classifier for diabetes prediction," *IAES Int. J. Artif. Intell.*, vol. 13, no. 2, pp. 1567–1573, 2024, doi: 10.11591/ijai.v13.i2.pp1567-1573.

[9] A. Zamsuri, S. Defit, and G. W. Nurcahyo, "Classification Of Multiple Emotions In Indonesian Text Using The K-Nearest Neighbor Method," *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 2, pp. 1012–1021, 2023, doi: 10.37385/jaets.v4i2.1964.

[10] M. Zhu, J. Lin, G. Cao, J. Zhang, X. Zhang, J. Zhou, and Y. Gao, "Prediction of constitutive model for basalt fiber reinforced concrete based on PSO-KNN," *Heliyon*, vol. 10, no. 11, pp. 1-15, 2024. doi: 10.1016/j.heliyon.2024.e32240.

[11] A. M. Elmogy, U. Tariq, A. Ibrahim, and A. Mohammed, "Fake Reviews Detection using Supervised Machine Learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 601–606, 2021, doi: 10.14569/IJACSA.2021.0120169.

[12] A. J. Barid, Hadiyanto, and A. Wibowo, "Optimization of the algorithms use ensemble and synthetic minority oversampling technique for air quality classification," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 33, no. 3, pp. 1632–1640, 2024, doi: 10.11591/ijeecs.v33.i3.pp1632-1640.

[13] A. Chachoui, N. Azizi, R. Hotte, and T. Bensebaa, "Enhancing algorithmic assessment in education: Equi-fused-data-based SMOTE for balanced learning," *Computers and Education: Artificial Intelligence*, vol. 6, no. June, 2024. doi: 10.1016/j.caeai.2024.100222.

[14] C. Supriyanto, F. A. Rafrastara, A. Amiral, "Malware Detection Using K-Nearest Neighbor Algorithm and Feature Selection," *J. Media Inform. Budidarma*, vol. 8, no. 1, pp. 412–420, 2024, doi: 10.30865/mib.v8i1.6970.

[15] S. Tomar, D. Dembla, and Y. Chaba, "Analysis and Enhancement of Prediction of Cardiovascular Disease Diagnosis using Machine Learning Models SVM, SGD, and XGBoost," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 4, pp. 469–479, 2024, doi: 10.14569/IJACSA.2024.0150449.

[16] A. Mishra, S. Mishra, and P. Jain, "Malware Category Prediction Using KNN And SVM Classifiers," *Int. J. Mech. Eng. Technol.*, vol. 10, no. 02, pp. 787–797, 2019.

[17] W. Chimphlee and S. Chimphlee, "Hyperparameters optimization XGBoost for network intrusion detection using CSE-CIC-IDS 2018 dataset," *IAES Int. J. Artif. Intell.*, vol. 13, no. 1, pp. 817–826, 2024, doi: 10.11591/ijai.v13.i1.pp817-826.

[18] Nur Ghaniaviyanto Ramadhan, “Indonesian Online News Topics Classification using Word2Vec and K-Nearest Neighbor,” *J. RESTI*, vol. 5, no. 6, pp. 1083–1089, 2021, doi: 10.29207/resti.v5i6.3547.

[19] D. Daimari, S. Mondal, B. Brahma, and A. Nag, “Favorite Book Prediction System Using Machine Learning Algorithms,” *J. Appl. Eng. Technol. Sci.*, vol. 4, no. 2, pp. 983–991, 2023, doi: 10.37385/jaets.v4i2.1925.

[20] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, “Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 874, no. 1, pp. 0–6, 2020, doi: 10.1088/1757-899X/874/1/012017.

[21] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, “The influence of preprocessing on text classification using a bag-of-words representation,” *PLoS One*, vol. 15, no. 5, pp. 1–22, 2020, doi: 10.1371/journal.pone.0232525.

[22] M. Novo-Lourés, R. Pavón, R. Laza, D. Ruano-Ordas, and J. R. Méndez, “Using natural language preprocessing architecture (NLPA) for big data text sources,” *Sci. Program.*, vol. 2020, no. jan, pp. 1-13, 2020, doi: 10.1155/2020/2390941.

[23] W. Bourequat and H. Mourad, “Sentiment Analysis Approach for Analyzing iPhone Release using Support Vector Machine,” *Int. J. Adv. Data Inf. Syst.*, vol. 2, no. 1, pp. 36–44, 2021, doi: 10.25008/ijadis.v2i1.1216.

[24] B. Rodrawangpai and W. Daungjaiboon, "Improving text classification with transformers and layer normalization," *Machine Learning with Applications*, vol. 10, no. December, pp. 1-9, 2022. doi: 10.1016/j.mlwa.2022.100403.

[25] N. Garg and K. Sharma, “Text pre-processing of multilingual for sentiment analysis based on social network data,” *Int. J. Electr. Comput. Eng.*, vol. 12, no. 1, pp. 776–784, 2022, doi: 10.11591/ijece.v12i1.pp776-784.

[26] L. Hickman, S. Thapa, L. Tay, M. Cao, and P. Srinivasan, “Text Preprocessing for Text Mining in Organizational Research: Review and Recommendations,” *Organ. Res. Methods*, vol. 25, no. 1, pp. 114–146, 2022, doi: 10.1177/1094428120971683.

[27] R. Kalaivani and R. Marivendan, “The effect of stop word removal and stemming in datapreprocessing,” *Ann. R.S.C.B*, vol. 25, no. 6, pp. 739–746, 2021.

[28] B. Zhu and W. Pan, "Chinese text classification method based on sentence information enhancement and feature fusion," *Heliyon*, vol. 10, no. 17, pp. 1-10, 2024. doi: 10.1016/j.heliyon.2024.e36861.

[29] J. Zhou, Z. Ye, S. Zhang, Z. Geng, N. Han, and T. Yang, "Investigating response behavior through TF-IDF and Word2vec text analysis: A case study of PISA 2012 problem-solving process data," *Heliyon*, vol. 10, no. 16, pp. 1-22, 2024. doi: 10.1016/j.heliyon.2024.e35945.

[30] T. Colibazzi et al., “Identifying Splitting Through Sentiment Analysis,” *J. Pers. Disord.*, vol. 37, no. 1, pp. 36–48, 2023, doi: 10.1521/pedi.2023.37.1.36.

[31] A. Febriani, R. Wahyuni, Y. Irawan, and R. Melyanti, “Improved Hybrid Machine and Deep Learning Model for Optimization of Smart Egg Incubator,” *J. Appl. Data Sci.*, vol. 5, no. 3, pp. 1052–1068, 2024.

[32] M. Rezwanul, A. Ali, and A. Rahman, “Sentiment Analysis on Twitter Data using KNN and SVM,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 6, pp. 19–25, 2017, doi: 10.14569/ijacsa.2017.080603.

[33] S. Wang, Y. Dai, J. Shen, and J. Xuan, “Research on expansion and classification of imbalanced data based on SMOTE algorithm,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–11, 2021, doi: 10.1038/s41598-021-03430-5.

[34] Y. Ding, H. Zhu, R. Chen, and R. Li, “An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification,” *Appl. Sci.*, vol. 12, no. 12, pp. 1-13, 2022, doi: 10.3390/app12125872.

[35] J. Xu, Y. Jiang, and C. Yang, “Landslide Displacement Prediction during the Sliding Process Using XGBoost, SVR and RNNs,” *Appl. Sci.*, vol. 12, no. 12, pp. 1-16, 2022, doi: 10.3390/app12126056.

[36] Y. Irawan, "Moving Load Robot Using Wifi Network and Android Based," *Journal of Robotics and Control (JRC)*, vol. 2, no. 3, pp. 217-221, 2021.

[37] A. G. Pertiwi, N. Bachtiar, R. Kusumaningrum, I. Waspada, and A. Wibowo, “Comparison of performance of k-nearest neighbor algorithm using smote and k-nearest neighbor algorithm without smote in diagnosis of diabetes disease in balanced data,” *J. Phys. Conf. Ser.*, vol. 1524, no. 1, pp. 1–8, 2020, doi: 10.1088/1742-6596/1524/1/012048.